Ornithokrites

Lukasz Tracewski
February 23, 2014

1 OVERVIEW

Ornithokrites is a transliteration of ancient Greek ὀρνϊθοκρίτης, meaning interpreter of flight or cries of birds. With its rather ambitious name, the program itself is a tool meant for the automatic identification of kiwi calls from audio recordings. It is designed to cope with large variations of environmental conditions and low quality of input data. For each provided audio file, the program tries to find whether it contains any kiwi calls and, if so, whether they are male, female or both.
Complete source code can be found on project's web site: `https://github.com/tracek/Ornithokrites`.

2 HOW TO USE IT

Expected input are monaural (single-channel) audio files in Waveform Audio File Format (commonly known as WAV or WAVE). Following sections explain two ways of running the program: user-friendly 2.1 and user-hostile 2.2.

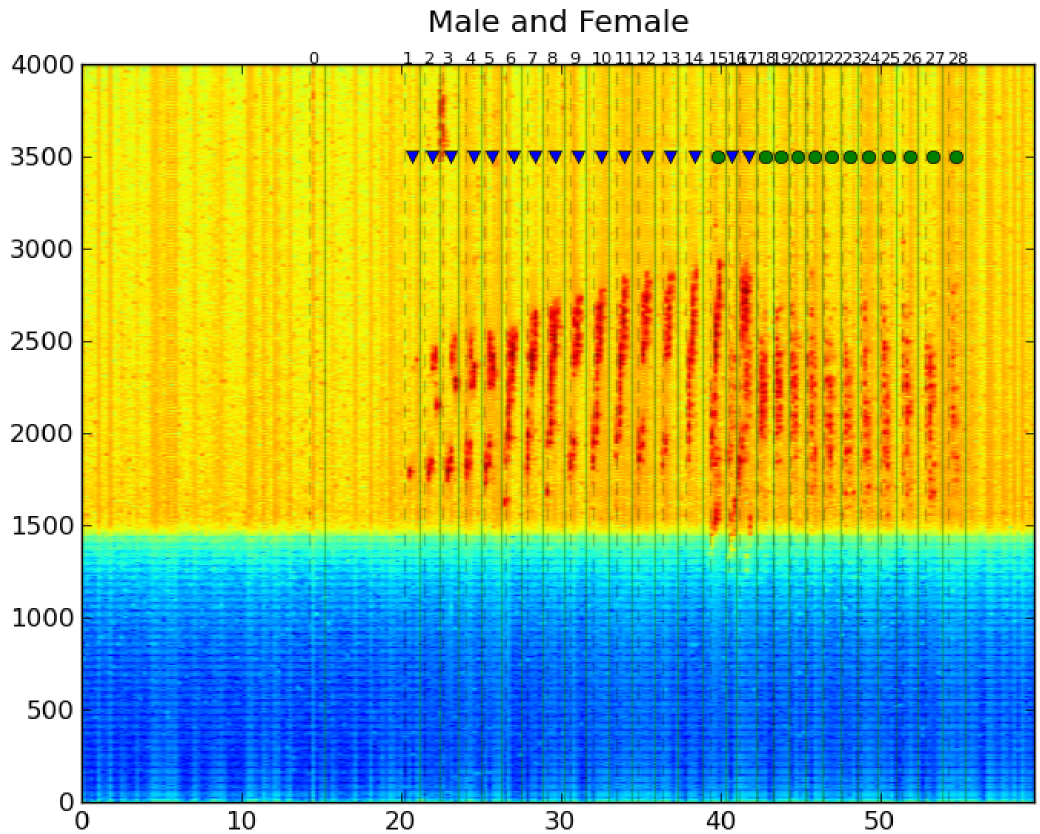
2.1 WEB INTERFACE

If the data is stored on Amazon Web Services S3 bucket, then by far easiest way of using the program is through a password-protected web site: `http://kiwi-finder.info`. The protection is needed since only one user at a time can run the program.
After providing the credentials user is directed to a simple web form that serves as an interface to the application.

- Bucket name: name of Amazon Web Services S3 bucket, e.g. *kiwicalldata*.

- Execute: connect to data store, download the recordings and run kiwi calls identification. It is a long-lasting operation. Closing the web page does not stop execution.
- Report: show results. Since they are generated live, user can click the button at any moment to get current state of affairs. Only text is printed, making it very fast.
- Show details: show detailed results. In this mode additional data is provided: spectrogram 2.1 with identified Regions Of Interest: blue triangles are detected male kiwi calls and green circles stand for female kiwi calls. Mind that single detected kiwi call does not make a kiwi; only in groups program considers them as candidates. Additionally, an option to play the original audio is provided, allowing user to verify program's predictions.
- Clear: stop execution of the program and clear all intermediate results.

Figure 2.1: Example of spectrogram showing male and female kiwi



2.2 INTERACTIVE MODE

The program is written in Python, which means running it directly, either from command line or in interactive mode, requires installation of all dependent modules; complete list can be found on project's page. Due to the dependencies installing is not an easy task.

2.2.1 Batch mode - command line

Executing program with '-h' switch, i.e. `ornithiokrites.py -h`, will print complete help with command line arguments explained. This mode allows batch processing of files stored on a disc.

2.2.2 Single-file mode - graphical user interface

If no command line arguments are provided then program will start in interactive mode. With open file dialogue user can select a single file for analysis.

3 HOW IT WORKS

After the recordings are ready following steps take place:

1. **Apply high-pass filter**. This step will reduce strength of any signal below 1500 Hz. Experiments so far have shown that kiwi rarely show any vocalization below this value. It also helps to eliminate bird calls of no interest to us, e.g. long-tailed cuckoo.

2. **Find Regions of Interest** (ROIs), defined as any signal different than background noise. Since length of a single kiwi call is roughly constant, ROI length is fixed to one second. First onsets are found by calculating local energy of the input spectral frame and taking those above certain dynamically-assessed threshold. Then from the detected onset a delay of -0.2s is taken to compensate for possible discontinuities. End of ROI is defined as $+0.8$s after beginning of the onset, summing to 1s interval. The algorithm is made sensitive, since potential cost of not including kiwi candidate in a set of ROIs is much higher then adding noise-only ROI.

3. **Reduce noise**. Since ROIs are identified, Noise-Only Regions (NORs) can be estimated as anything outside ROIs plus some margin. Based on NORs spectral subtraction is performed: knowing noise spectrum we can try to eliminate noise over whole sample.

4. **Calculate Audio Features**. Those features will serve as a kiwi audio signature, allowing to discriminate kiwi male from female - and the two from not a kiwi. For each ROI following features are calculated:
   - spectral flatness
   - perceptual spread
   - spectral roll-off
   - spectral decrease

- spectral shape statistics
- spectral slope
- Linear Predictive Coding (LPC)
- Line Spectral Pairs (LSP)

AFs are calculated with `Yaafe` library. On its project page `http://yaafe.sourceforge.net/features.html` a complete description of above-mentioned features can be found.

5. **Perform kiwi identification**. At this stage Audio Features are extracted from the recording. Based on those, a Machine Learning algorithm, that is Support Vector Machine (SVM), will try to classify ROI as kiwi male, kiwi female and not a kiwi. Additional rules are then applied, employing our knowledge on repetitive character of kiwi calls. Only once sufficiently long set of calls is identified, the kiwi presence is marked.

6. **Report**. Algorithm output can be: female, male, male and female and no kiwi detected.

4 VALIDATION RESULTS

Program was tested using stratified 5-fold cross-validation [1]. Per Region Of Interest model got 93% accuracy in telling kiwi apart from not a kiwi and 90% overall. Per file those numbers raise to 99% and 97% respectively, which is possible thanks to employment of the most prominent feature of kiwi calls: their repetitive character.

Table 4.1: Confusion matrix

	Kiwi Male	Kiwi Female	Male and Female	Not a kiwi
Kiwi Male	65	0	0	1
Kiwi Female	0	35	0	0
Male and Female	1	3	26	0
Not a kiwi	0	0	0	30

As can be seen from the confusion matrix [2] in table 4, the program may misclassify a file when male and female are chatting label the pair as either Male or Female.

Since the program was trained on 1-minute samples, it very well might be that many times longer recordings will require a bit different approach with respect to noise reduction - and thus prove more challenging to the algorithm; nothing that cannot be fixed with more training data.

Processing of a single, one-minute file takes roughly 2 seconds on a single core of Intel Core-i5 @ 3.40 GHz.

[1] `http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29`
[2] `http://en.wikipedia.org/wiki/Confusion_matrix`

5 Developer's notes

Following notes provide more technical information of little interest for the end user.

5.1 Technology

Ornithokrites is written in Python 2.7, with all of its heavy-duty numerical computations done in C/C++ compiled libraries. Comments in the code were formatted according to NumPy guidelines, so that they can be easily extracted by a capable program (e.g. Spyder). Program was tested on Linux, although, given enough determination, it should be also possible to run it on Windows and Mac OS.

5.2 Machine Learning algorithm

The Machine Learning algorithm used in this program, that is Support Vector Machine with Gaussian kernel, belongs to a class of supervised models. As a consequence, it requires labelled training set for fitting the model. At this moment 3318 fragments were classified as either male, female or not a kiwi. It is a manual task with a labelling dependant on ones proficiency in kiwi identification - and hence prone to errors.

6 Acknowledgments

I wish to thank Barry Polley for providing this utmost interesting challenge and his assistance along the way. My great appreciation also goes to Pat Miller of `birdingNZ.net` community that supplied me with invaluable help concerning kiwi identification; without his aid I would not be able to achieve such high accuracy.

7 Contact information

My e-mail: lukasz.tracewski@gmail.com

Project's web page: `https://github.com/tracek/Ornithokrites`.